09/512,620 Patent #8

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

4/24/03
Not Entered

In re Application of: Harlan SEXTON et al.

Application No.: 09/512,620

Filed: February 25, 2000

Attorney Docket No.: 50277-0403
Client Docket No.: OID-1997-048-19

Reply under 37 CFR 1.116
EXPEDITED PROCEDURE

Examiner: Newgen, L.

Group Art Unit: 2127

For: USING A VIRTUAL MACHINE INSTANCE AS THE BASIC UNIT OF USER EXECUTION IN A SERVER ENVIRONMENT

BOX AF
Assistant Commissioner for Patents
Washington, D.C. 20231

RESPONSE UNDER 37 CFR 1.116

Enter upon appeal 4/25/03

Dear Sir:

This is in response to the final Office Action of April 3, 2003, in which claims 1-16 are pending.

The final Office Action mailed April 3, 2003 rejected claims 1 and 9 as obvious under 35 U.S.C. § 103 based on *Bayeh* (US 6,223,202) in view of *Johnson et al.* (US 6,330,709), claims 2-5 and 10-13 over *Bayeh* and *Johnson et al.* in view of *Bugnion et al.* (US 6,975,938), claims 6 and 14 over *Bayeh* and *Johnson et al.* in view of *Miner et al.* (US 6,047,053), claims 7 and 15 over *Bayeh* and *Johnson et al.* in view of *Loomans* (US 6,393,605), and claims 8 and 16 over *Bayeh* and *Johnson et al.* in view of *Heiney et al.* (US 6,401,109).

The rejection of claims under §103 is respectfully traversed because neither *Bayeh* nor *Johnson et al.* teach or otherwise the limitations of the claims. For example, independent claim 1 recites:

> 1. A method for servicing requests received by a server in a multiple-user environment, the method comprising the steps of:
> establishing a first session between said server and a first user;
> establishing a second session between said server and a second user;
> responding to requests that are received by said server in said first session by executing virtual machine code using a first virtual machine instance; and
> responding to requests that are received by said server in said second session by executing virtual machine code using a second virtual machine instance;
> wherein said first virtual machine instance and said second virtual machine instance are distinct instances of a same type of virtual machine;
> wherein said first virtual machine instance exists within said server concurrently with said second virtual machine instance; and
> wherein said first virtual machine instance and said second virtual machine instance are two **of a plurality of virtual machine instances**, associated with said server, that **share access to data stored in a shared state area** allocated in volatile memory associated with said server.

Claim 9 also recites "**two of a plurality of virtual machine instances ...that share access to data stored in a shared state area** allocated in volatile memory." The Office Action correctly acknowledges that *Bayeh* fails to explicitly teach this limitation. However, this limitation is also not found in *Johnson et al.* *Johnson et al.* is directed to several objects used to implement persistent objects in a shared address space. (col. 3:41-43) and discloses that "operations in Java are performed by one object calling a method on another object. These objects can reside locally on the same machine or on separate JVMs physically located on separate computers or system." (col. 6:15-19). The Office Action states that "this indicates that objects on separate JVMs can also be shared as well." (Page 9) This inference is contradicted by *Johnson et al.* itself: "there are no sharing of static variables among JVMs." (col. 18:62-63). Furthermore, *Johnson et al.* does not disclose any details on how an object can call a method on an object on a separate computer. Presumably, the remote call would be accomplished in a conventional manner, i.e. with a remote procedure call in which a copy of the object is transmitted not by using "**access to data stored in a shared state area.**"

Furthermore, the use of the shared address space (SAS) is only disclosed in *Johnson et al.* to be between a JVM and an application (col. 20:7-20), not between a "**plurality of virtual machine instances"** as recited in the claims. *Johnson et al.* explicitly teaches against this feature: "different JVMs do not access the same static variables. Each JVM has its own copies of any static variables defined for a class of which it has an instance." (col. 18:42-44).

Independent claims 1 and 9 also recite:

> wherein said first virtual machine instance and said second virtual machine instance are two of a plurality of virtual machine instances, associated with said server, that share access to data stored **in a shared state area allocated in volatile memory** associated with said server.

The Office Action correctly acknowledges that *Bayeh* fails to explicitly teach this limitation. However, this limitation is also not found in *Johnson et al.*, because *Johnson et al.* discloses that the "JVM can store objects in **either** temporary local storage **or** in **permanent** SAS storage **220**" (col. 20:9-10). The temporary local storage is not shared, and the SAS is not in volatile memory. In fact *Johnson et al.* states: "in this application the term **shared address space** refers to the large address space that allows applications to store **persistent** data using single level store semantics." (col. 12:1-4). Therefore, disclosure of storing in a shared, permanent SAS storage **220** does not amount to a teaching of (and, in fact, teaches against) "**a shared state area allocated in volatile memory**."

Dependent claims 2-8 and 10-16 are allowable for at least the same reasons as their independent claims and are separately patentable on their own merits.

The "Official Notices" invoked appear irrelevant to the claim language. For example, in regard to "the sharing of application programs," (Page 4) the claims do not recite "application programs."

Therefore, the present application, as amended, overcomes the objections and rejections of record and is in condition for allowance. Favorable consideration is respectfully requested. If any unresolved issues remain, it is respectfully requested that the Examiner telephone the undersigned attorney at 703-425-8501 so that such issues may be resolved as expeditiously as possible.

Respectfully Submitted,

DITTHAVONG & CARLSON, P.C.

4.11.03
Date

Leila Abdi
Attorney/Agent for Applicant(s)
Reg. No. 52399

10507 Braddock Rd
Suite A
Fairfax, VA 22032
Tel. 703-425-8501
Fax. 703-425-8518